

FIG.1

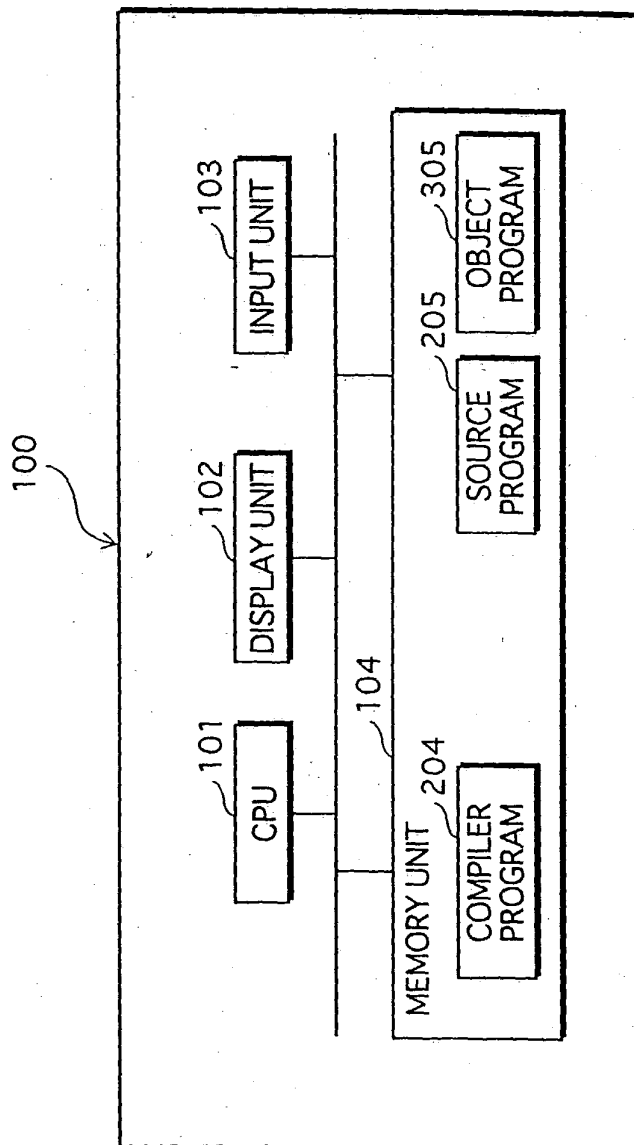


FIG.2

```
void dummy1(char);  
void dummy2(int);  
void dummy3(char*);  
void dummy4(double*);
```

```
100    void  
101    f(void)  
102    {  
        .  
        .  
        .  
110    char a;  
111    int b;  
112    char c[19];  
113    double d[4];  
        .  
        .  
        .  
140    dummy1(a);  
141    dummy2(b);  
142    dummy3(c);  
143    dummy4(d);  
        .  
        .  
        .  
199    return;  
200    }
```

FIG.3

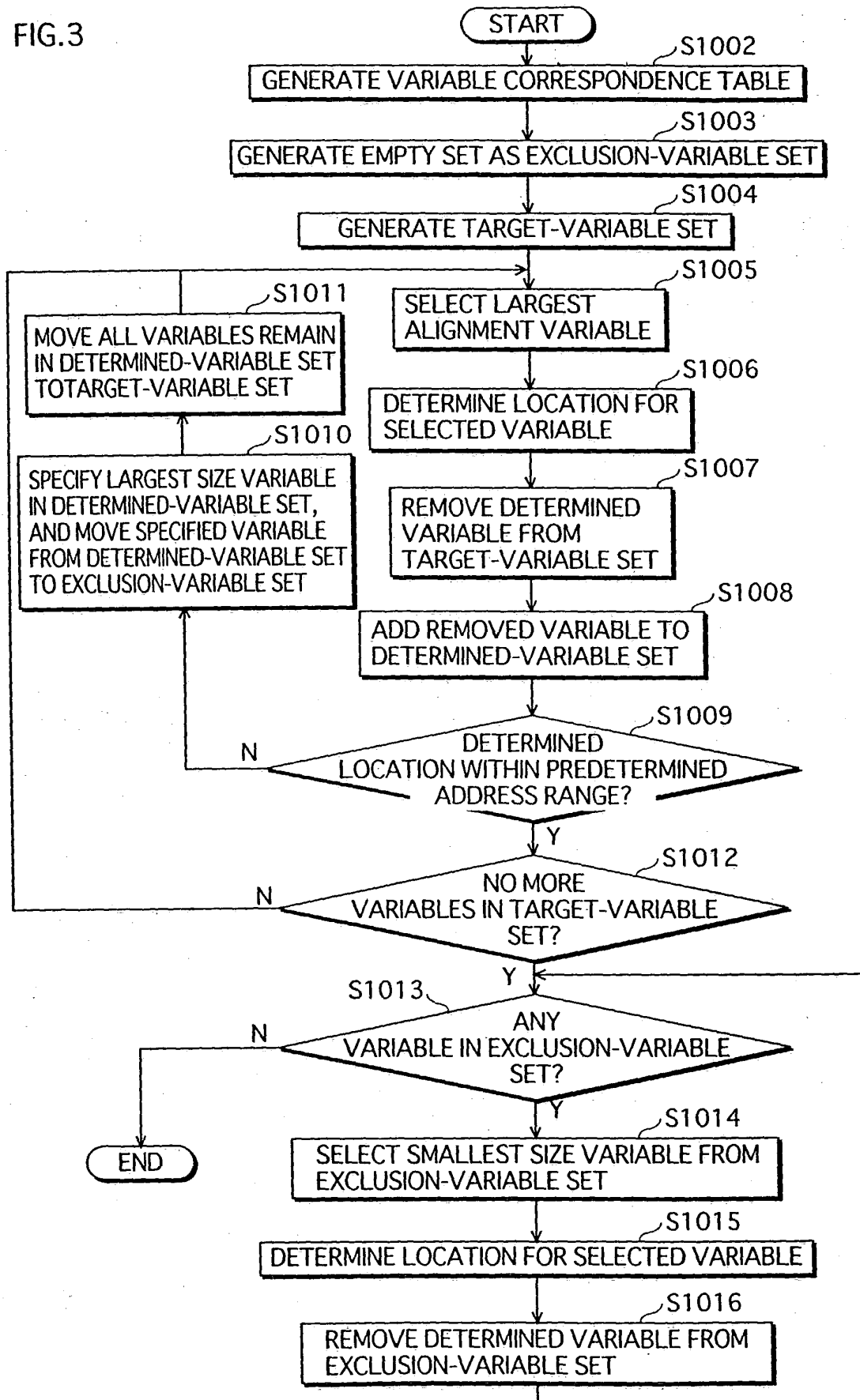


FIG.4

TYPE	VARIABLE NAME	SIZE	ALIGNMENT
char	a	1	1
int	b	4	4
char[]	c	19	1
double[]	d	32	8

FIG. 5

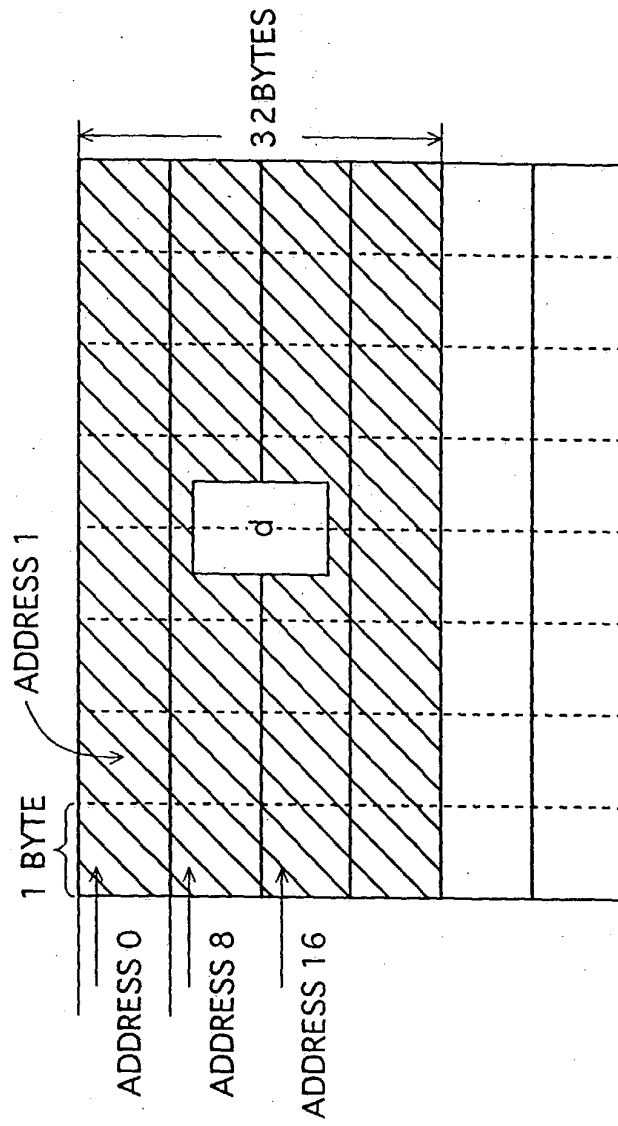


FIG. 6

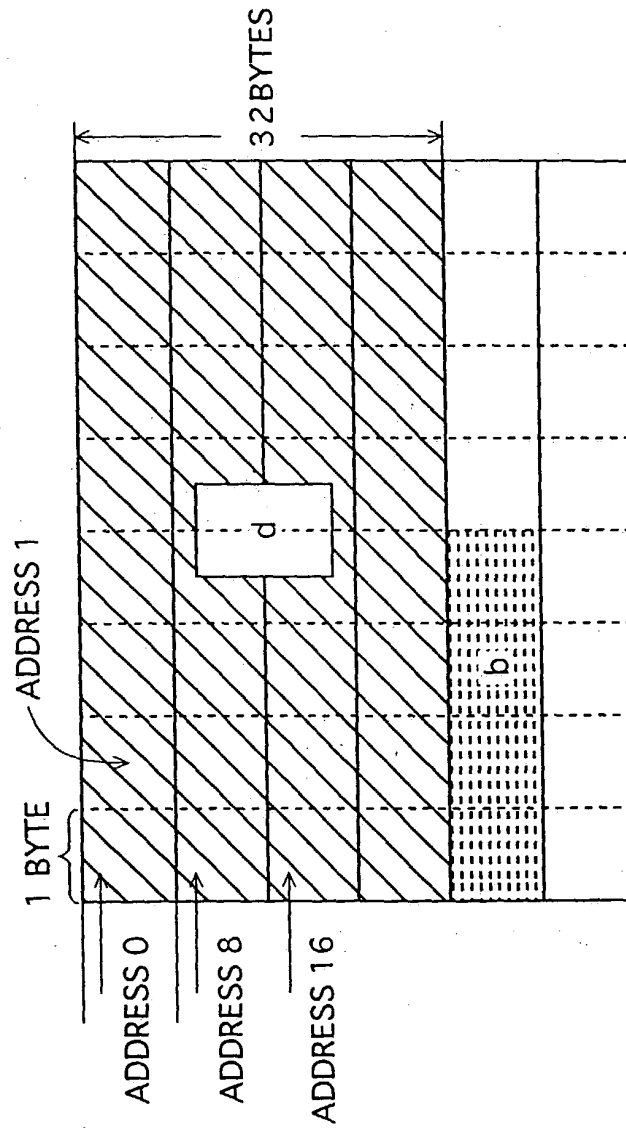


FIG. 7

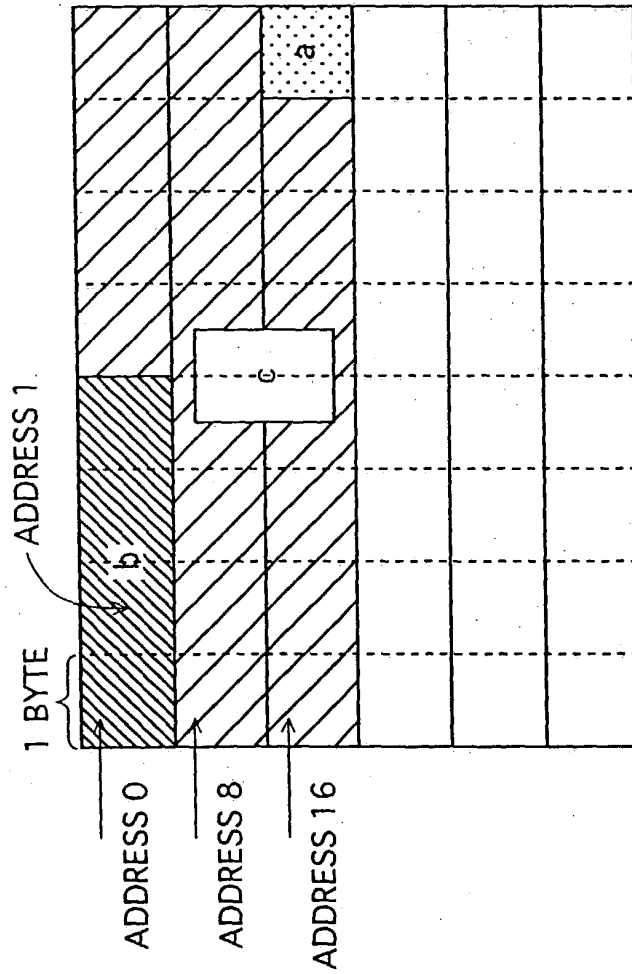


FIG. 8

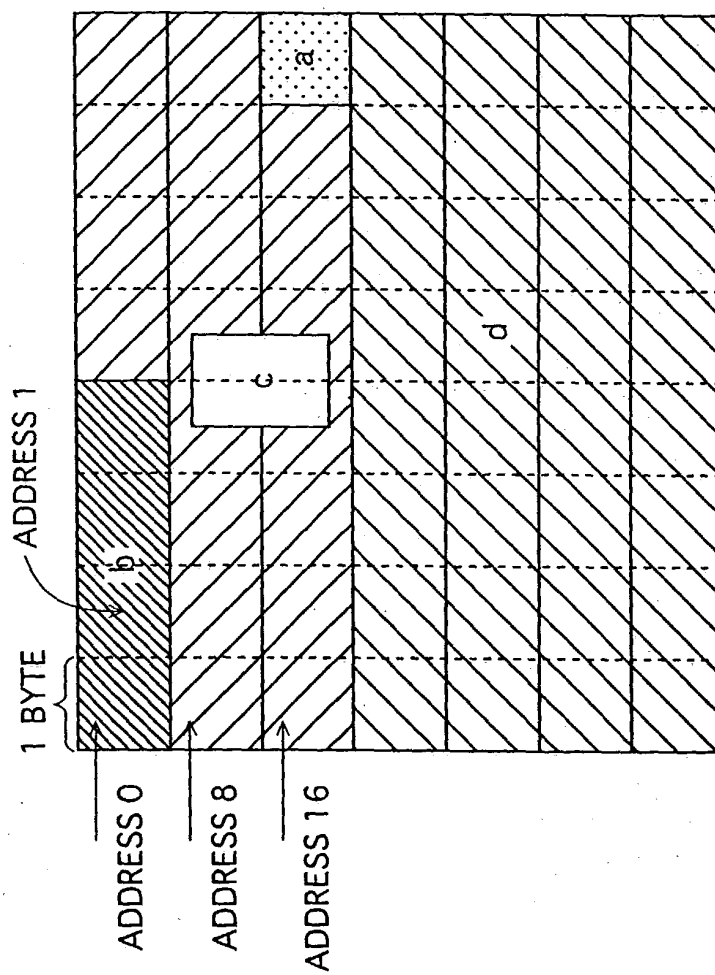


FIG.9

210 ld1 r0,(sp,23) //ACCESS TO a

•

•

•

220 ld4 r0,(sp,0) //ACCESS TO b

•

•

•

230 ld1 r0,(sp,4) //ACCESS TO c

•

•

•

240 ld8 r0,(sp,24) //ACCESS TO d

•

•

•

FIG.10

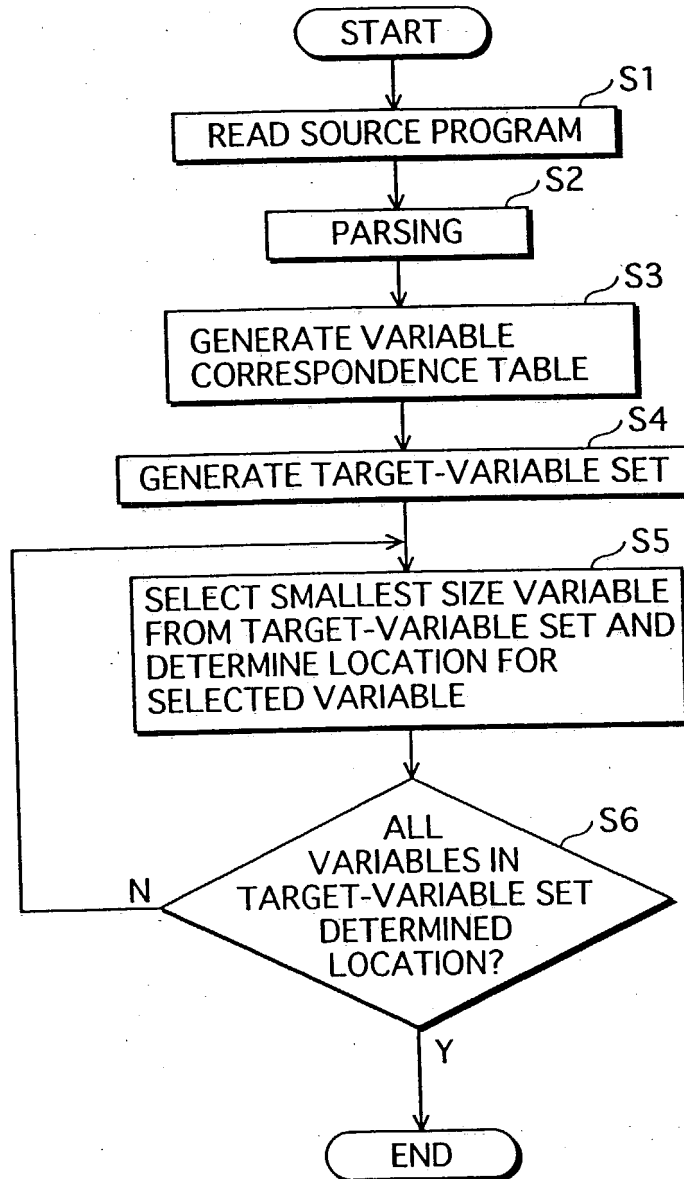
```
310    ld1 r0,(sp,0) //ACCESS TO a
      .
      .
      .
320    ld4 r0,(sp,4) //ACCESS TO b
      .
      .
      .
330    ld1 r0,(sp,8) //ACCESS TO c
      .
      .
      .
340    mov r1,32    //ACCESS TO d
350    ld8 r0,(r1)  //ACCESS TO d
      .
      .
      .
```

FIG.11

```
410      mov r1,55      //ACCESS TO a
420      ld1 r0,(r1)    //ACCESS TO a
      .
      .
      .
430      mov r1,32      //ACCESS TO b
440      ld4 r0,(r1)    //ACCESS TO b
      .
      .
      .
450      mov r1,36      //ACCESS TO c
460      ld1 r0,(r1)    //ACCESS TO c
      .
      .
      .
470      ld8 r0,(sp,0)  //ACCESS TO d
      .
      .
      .
```

FIG.12

Prior Art



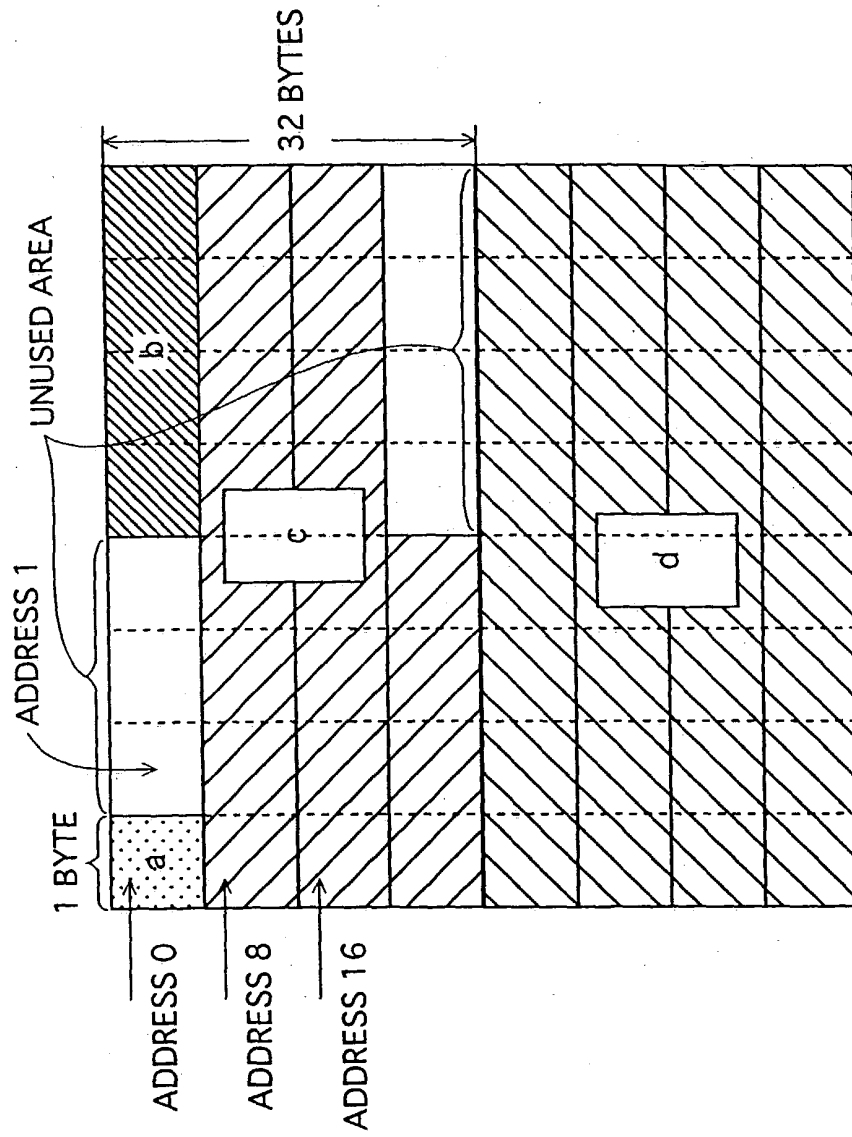
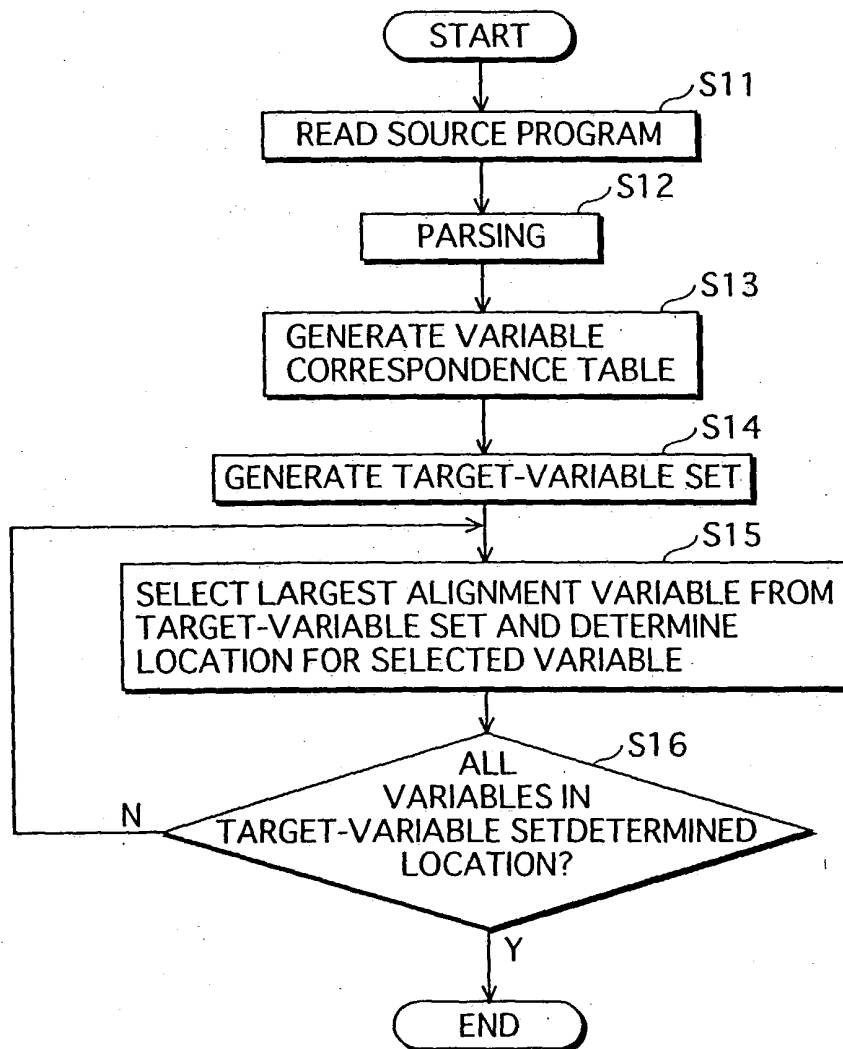


FIG.14

Prior Art



Prior Art

FIG.15

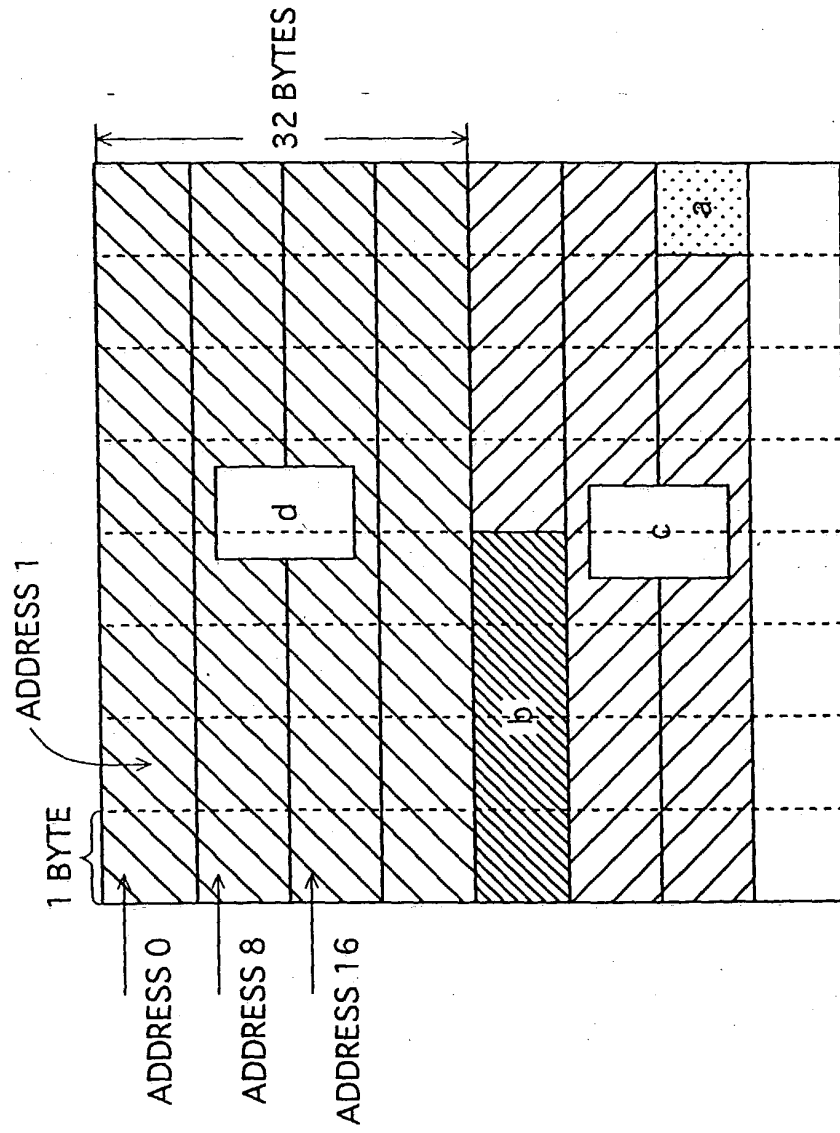


FIG.16

TYPE	VARIABLE NAME	SIZE	ALIGNMENT	REFERENCE FREQUENCY
char	a	1	1	5
int	b	4	4	45
char[]	c	19	1	7
double[]	d	32	8	4